A machine learning-based calibration of a 1D ejector model from CFD

36th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS)

25-30 June 2023

jan.vandenberghe@vki.ac.be

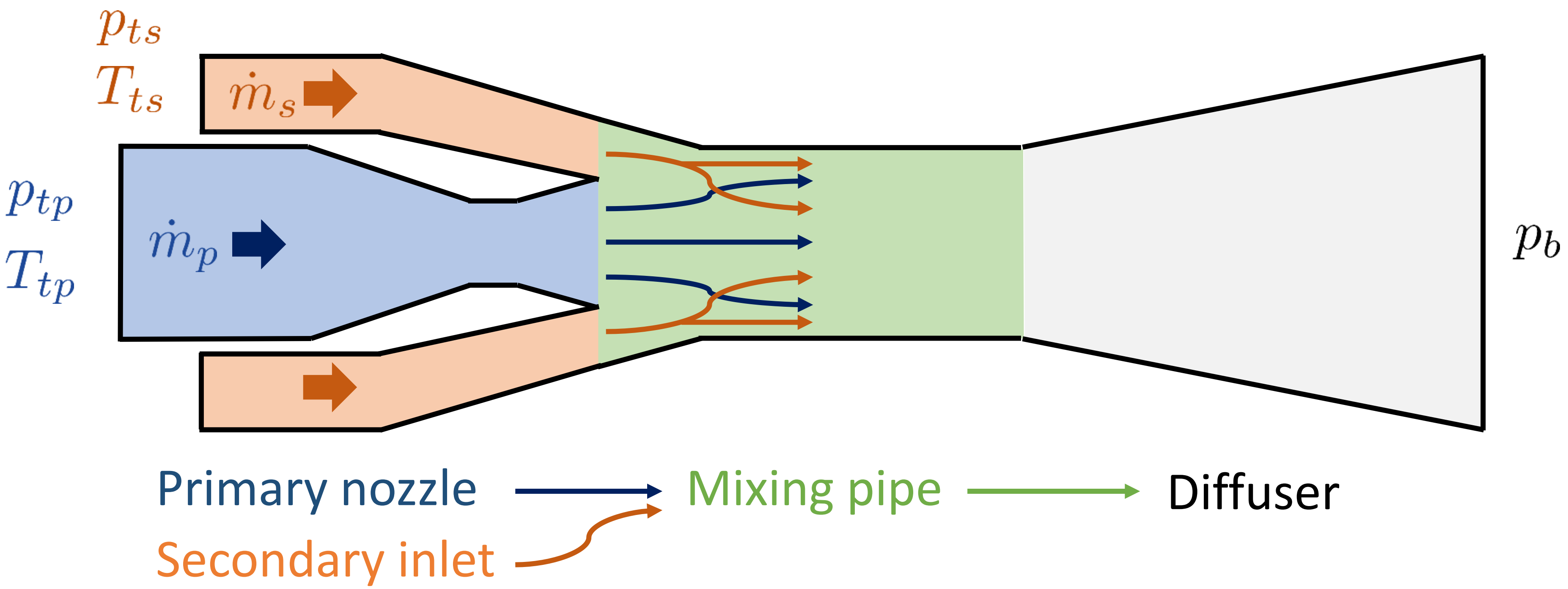
Jan Van den Berghe
Jagadish Babu Vemula
Yann Bartosiewicz
Miguel Alfonso Mendez

von KARMAN INSTITUTE FOR FLUID DYNAMICS

UCLouvain

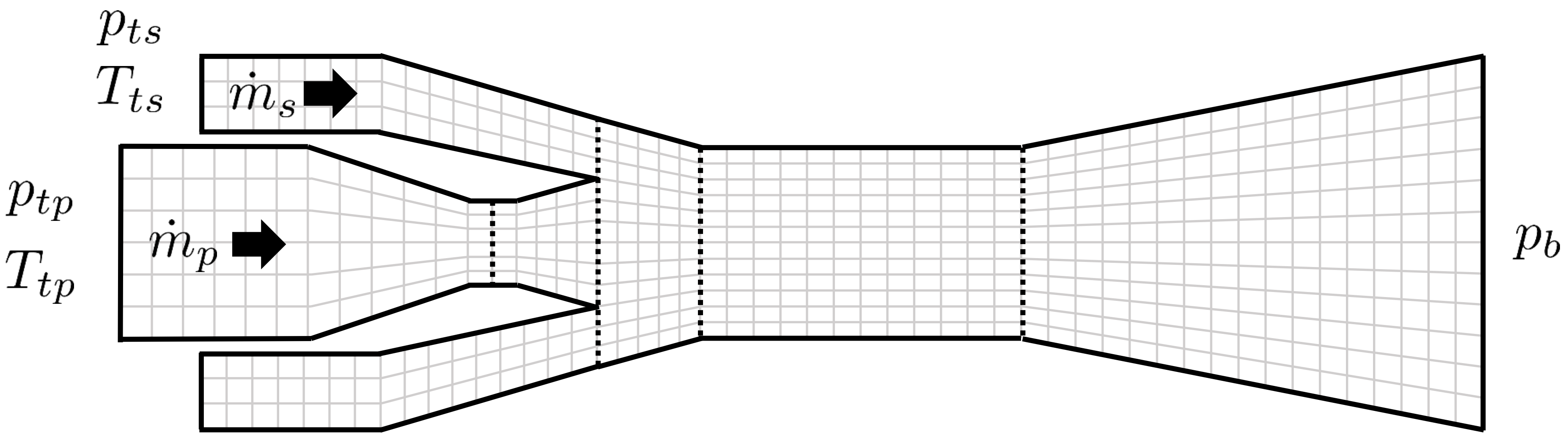
fnrs
LA LIBERTÉ DE CHERCHER

What is an ejector?



- Aeronautics (air)
- Refrigeration (CO_2)
- Desalination (steam)
- Process industry, power generation, …

What are the modelling options?



0D: global balances

- ✓ Fast
- ✗ Only global information
- ✗ Calibration

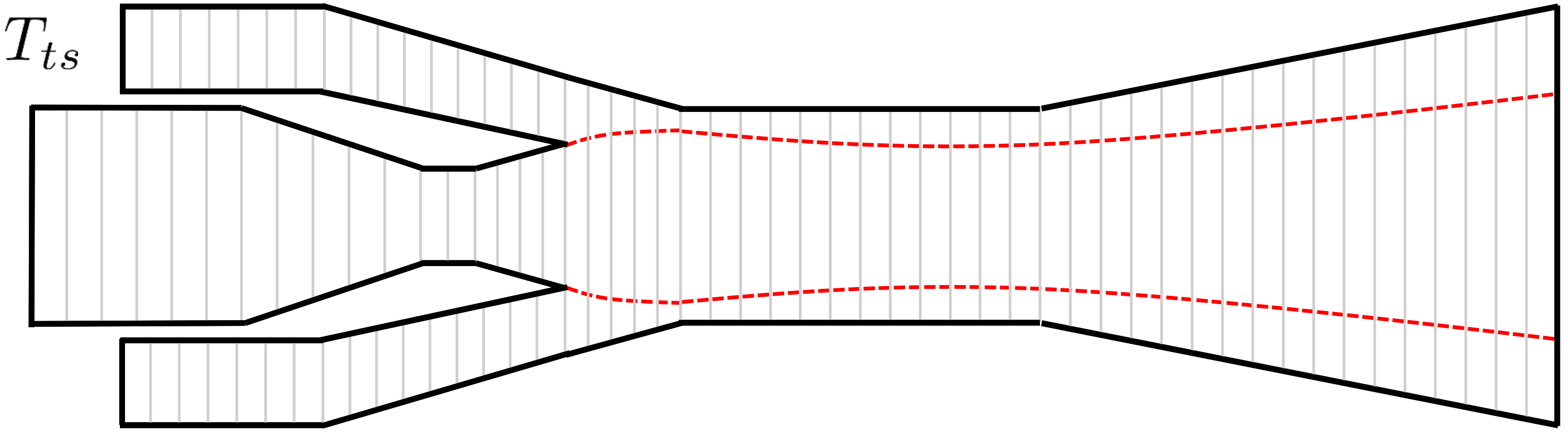
1D: a compromise

- ✓ Local information
- ✓ Fast
- ✗ Calibration (2D effects)
- ✗ Complex

2/3D: classic CFD

- ✓ Detailed flow field
- ✓ No calibration
- ✗ Computational cost
- ✗ Meshing and modelling

The 1D ejector model in this work



Conservation of mass, momentum and energy for each stream:

$$\begin{cases} d_x(\dot{m}_p) & = 0 \\ d_x(\dot{m}_p u_p) & = -A_p d_x p_p - \frac{1}{2} f_{ps} \left(\rho_p u_p^2 - \rho_s u_s^2\right) l_{ps} \\ d_x(\dot{m}_p h_{t,p}) & = 0 \end{cases}$$

$$\begin{cases} d_x(\dot{m}_s) & = 0 \\ d_x(\dot{m}_s u_s) & = -A_s d_x p_s + \underbrace{\frac{1}{2} f_{ps} \left(\rho_p u_p^2 - \rho_s u_s^2\right) l_{ps}}_{\text{Shear}} - \underbrace{\frac{1}{2} f_w (\rho_s u_s^2) l_w}_{\text{Wall friction}} \\ d_x(\dot{m}_s h_{t,s}) & = 0 \end{cases}$$

Constraints:

$$\begin{cases} p_p = p_s = p \\ A_p + A_s = A \end{cases}$$

Fluid: ideal gas

Which values of f_{ps}, f_w lead to a good match with experiments or CFD?

Contents

- **A general framework for machine learning-based calibration**
  - Problem statement
  - Physics-separated approach
  - Physics-integrated approach
- Application to a 1D ejector model
- Conclusion

A general framework for calibration

General formulation

Physical model

$$f\left(\boldsymbol{u}, \frac{d\boldsymbol{u}}{d\boldsymbol{x}}, \frac{d^2\boldsymbol{u}}{d\boldsymbol{x}^2}, \ldots, \boldsymbol{x}, \boldsymbol{p}\right) = 0$$

Closure relation

$$\boldsymbol{p} = g(\boldsymbol{u}, \boldsymbol{x}; \boldsymbol{w}) \qquad \boldsymbol{w} \in \mathbb{R}^{n_w}$$

Data (experiments, high-fidelity CFD)

$$[\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{x}}_i]$$

Calibration problem

Question: how to find the function g ?

➡ Parametrize g with weights $\boldsymbol{w}$ (linear, ANN, …)

"Physics-constrained machine learning"

Question: how to find the weights $\boldsymbol{w}$?

1D ejector model

Physical model

$$\boldsymbol{u} = [p, p_{tp}, p_{ts}, T_{tp}, T_{ts}, A_p, A_s]$$

$$\frac{d\boldsymbol{u}}{dx} = f(\boldsymbol{u}, \boldsymbol{p}) \qquad \boldsymbol{p} = [f_{ps}, f_w]$$

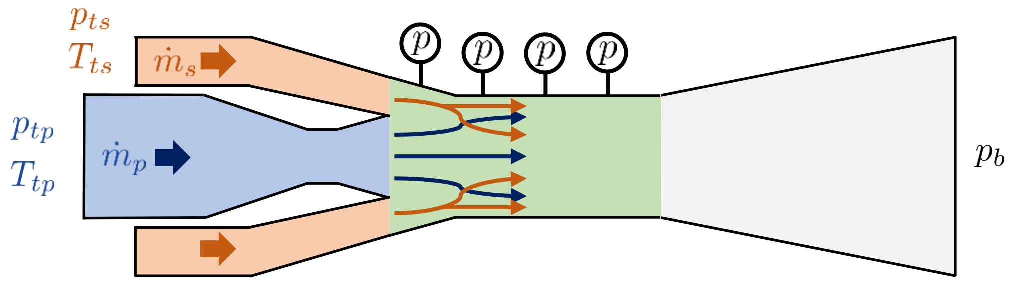
Closure relation

$$\boldsymbol{p} = [C_1, C_2]? \qquad \boldsymbol{p} = f(Re_p, Re_s, \ldots)?$$

Data

$$[\dot{m}_p, \dot{m}_s, p_{tp}, T_{tp}, p_{ts}, T_{ts}, p_b]$$

$$[p(x_1), p(x_2), \ldots, p(x_n)]$$



Physics-constrained machine learning for calibration

Physical model

$$f\left(\boldsymbol{u}, \frac{d\boldsymbol{u}}{d\boldsymbol{x}}, \frac{d^2\boldsymbol{u}}{d\boldsymbol{x}^2}, \ldots, \boldsymbol{x}, \boldsymbol{p}\right) = 0$$

Closure relation

$$\boldsymbol{p} = g(\boldsymbol{u}, \boldsymbol{x}; \boldsymbol{w}) \qquad \boldsymbol{w} \in \mathbb{R}^{n_w}$$

Data (experiments, high-fidelity CFD)

$$[\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{x}}_i]$$

Calibration problem

Question: how to find the weights $\boldsymbol{w}$?

Option 1: a set of reference parameters $\tilde{\boldsymbol{p}}$ are available:

$$J_1(\boldsymbol{w}) = \sum_i \frac{\left(\tilde{\boldsymbol{p}}_i - g(\boldsymbol{u}_i, \boldsymbol{x}_i; \boldsymbol{w})\right)^2}{\tilde{\boldsymbol{p}}_i^2}$$

The model is ***not*** involved: “physics-separated” approach

Option 2: penalize directly on the state

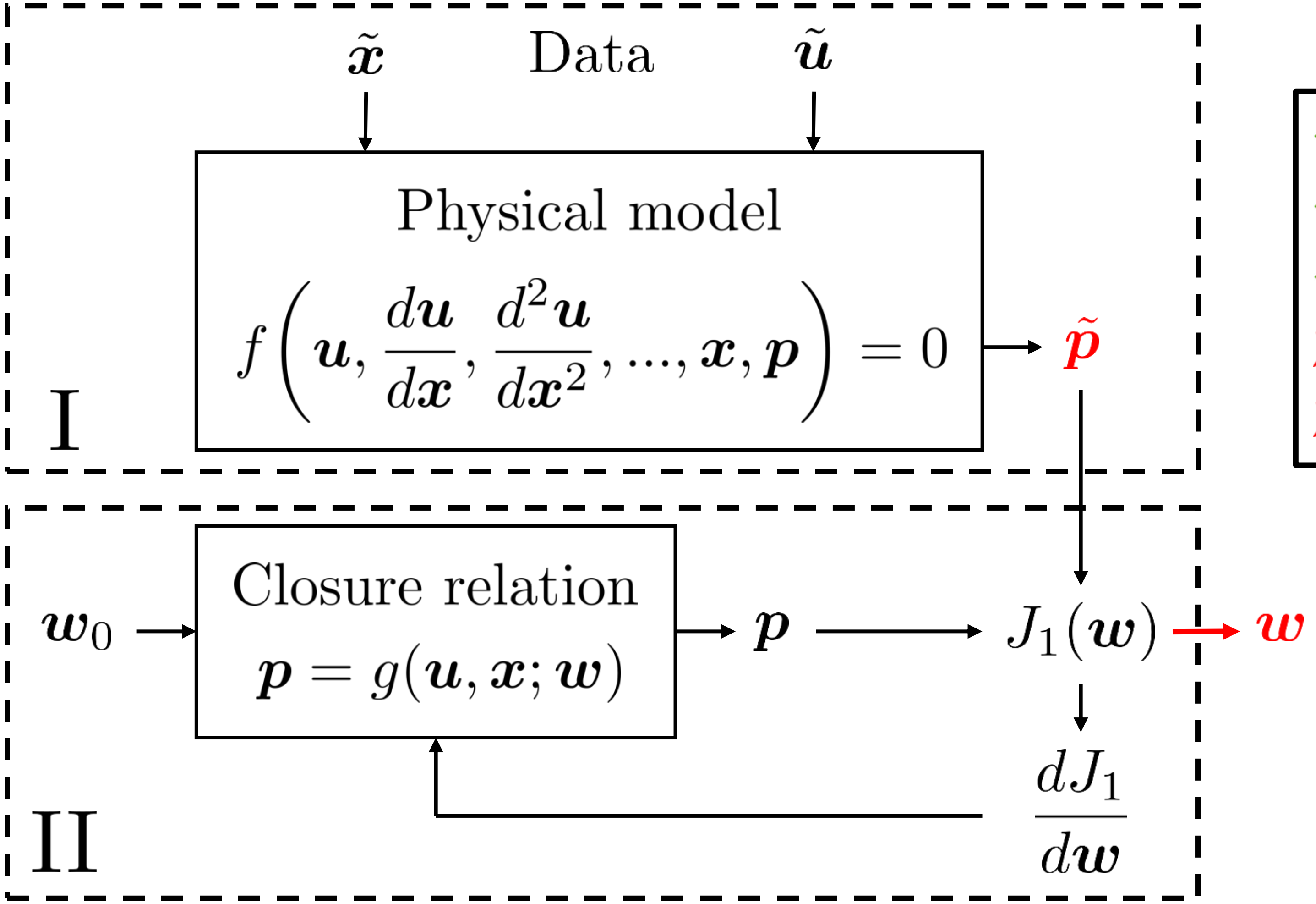
$$J_2(\boldsymbol{w}) = \sum_i \frac{\left(\tilde{\boldsymbol{u}}_i - \boldsymbol{u}_i(g(\boldsymbol{u}_i, \boldsymbol{x}_i; \boldsymbol{w}))\right)^2}{\tilde{\boldsymbol{u}}_i^2}$$

The model ***is*** involved: “physics-integrated” approach

Any derived variable from the state can be used,
at ***any*** position $\boldsymbol{x}$, by adapting J_2 (also sparse measurements).

Option 3: Combine the previous options

The physics-separated approach



- ✓ Reference parameters $\tilde{\boldsymbol{p}}$
- ✓ Easy to train
- ✓ Free choice of regressor
- ✗ Need dense data
- ✗ Unaware of the model

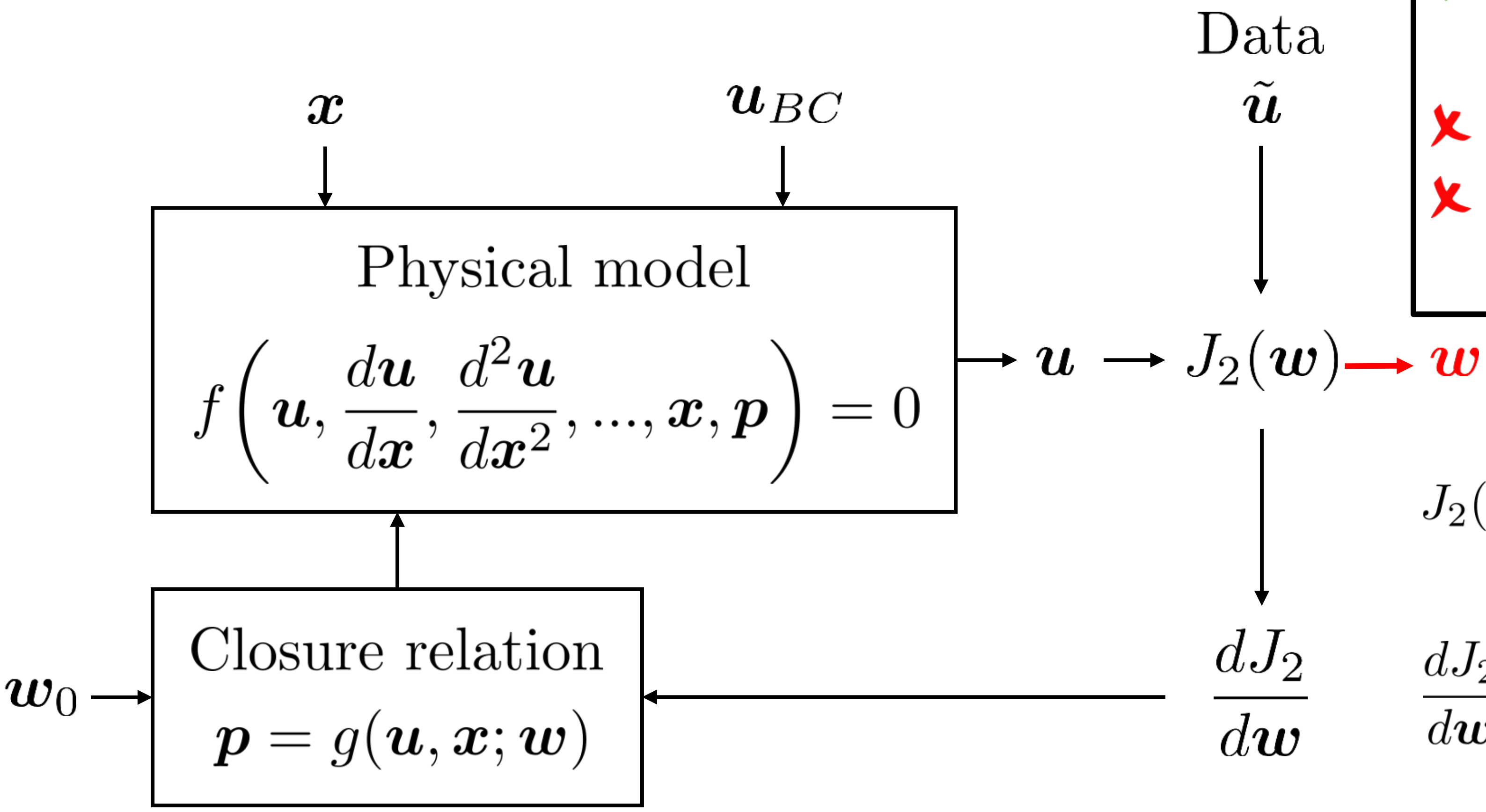
$$J_1(\boldsymbol{w}) = \sum_i \frac{\left(\tilde{\boldsymbol{p}}_i - g(\boldsymbol{u}_i, \boldsymbol{x}_i; \boldsymbol{w})\right)^2}{\tilde{\boldsymbol{p}}_i^2}$$

$$\frac{dJ_1}{d\boldsymbol{w}} = \frac{dJ_1}{d\boldsymbol{p}} \frac{dg}{d\boldsymbol{w}}$$

[1] Parish E. J., Duraisamy K., A paradigm for data-driven predictive modeling using field inversion and machine learning. Journal of computational physics (2016).

The physics-integrated approach

- ✓ Aware of the model
- ✓ Free choice of regressor
- ✓ Data from any (sparse) source, any quantity
- ✗ Potentially difficult to train
- ✗ More expensive and complex computation of the gradient



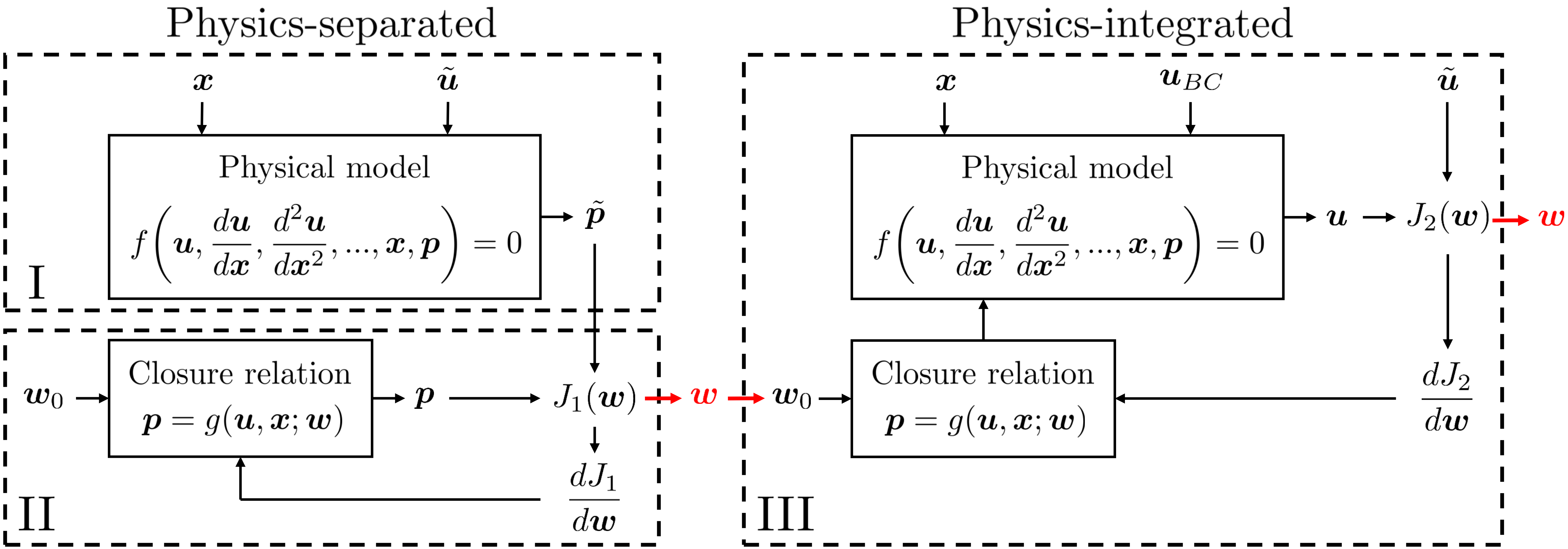
$$J_2(\boldsymbol{w}) = \sum_i \frac{\left(\tilde{\boldsymbol{u}}_i - \boldsymbol{u}_i(g(\boldsymbol{u}_i, \boldsymbol{x}_i; \boldsymbol{w}))\right)^2}{\tilde{\boldsymbol{u}}_i^2}$$

$$\frac{dJ_2}{d\boldsymbol{w}} = \frac{dJ_2}{d\boldsymbol{u}} \frac{d\boldsymbol{u}}{d\boldsymbol{p}} \frac{dg}{d\boldsymbol{w}}$$

Sensitivity of the model with respect to the parameters $\boldsymbol{p}$

[2] Holland J. R., Baeder J. D., Duraisamy K., Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training. AIAA Aviation Forum (2019).
[3] Sirignano J., MacArt J. F., Freund J. B., DPM: A deep learning PDE augmentation method with application to large-eddy simulation. Journal of Computational Physics (2020).
[4] MacArt J. F., Sirignano J., Freund J. B., Embedded training of neural-network subgrid-scale turbulence models. Physical Review Fluids (2021).

The two approaches can be combined

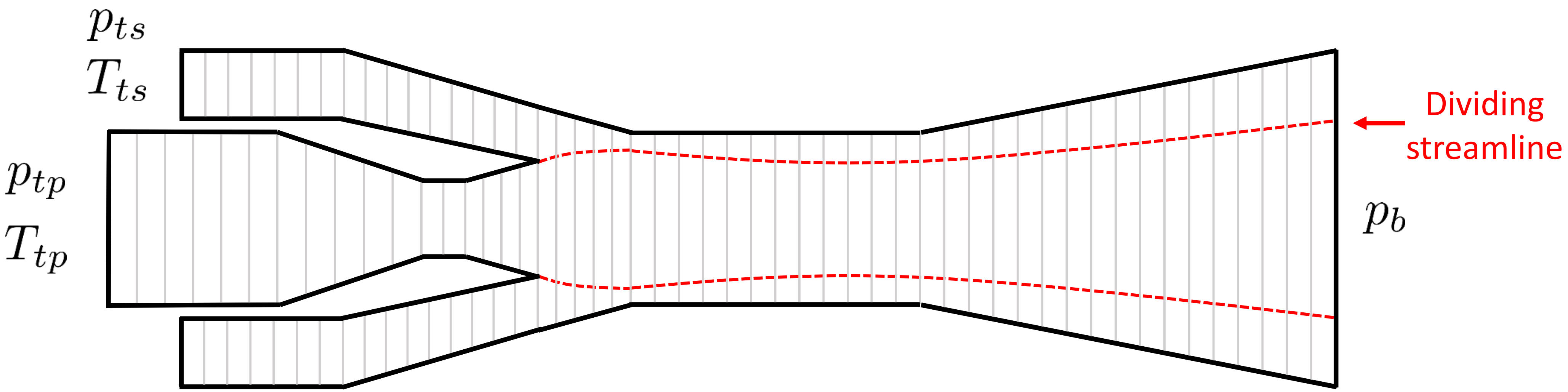


- ✓ Reference parameters (from step I)
- ✓ Aware of the model
- ✓ Free choice of regressor
- ✗ Need dense data

Contents

- A general framework for machine learning-based calibration
- **Application to a 1D ejector model**
  - **Working equations & reference data**
  - Physics-separated approach
  - Physics-integrated approach
- Conclusion

Practical working equations of the 1D ejector model



Desired form:

$$\boldsymbol{u} = \begin{pmatrix} p \\ p_{tp} \\ p_{ts} \\ T_{tp} \\ T_{ts} \\ A_p \\ A_s \end{pmatrix} \qquad \frac{d\boldsymbol{u}}{dx} = f(\boldsymbol{u}, \boldsymbol{p}) =$$

$$\boldsymbol{p} = \begin{pmatrix} f_{ps} \\ f_w \end{pmatrix}$$

Conservation of mass, momentum and energy for each stream:

$$\begin{cases} d_x(\dot{m}_p) &= 0 \\ d_x(\dot{m}_p u_p) &= -A_p d_x p_p - \frac{1}{2} f_{ps} \left(\rho_p u_p^2 - \rho_s u_s^2\right) l_{ps} \\ d_x(\dot{m}_p h_{t,p}) &= 0 \end{cases}$$

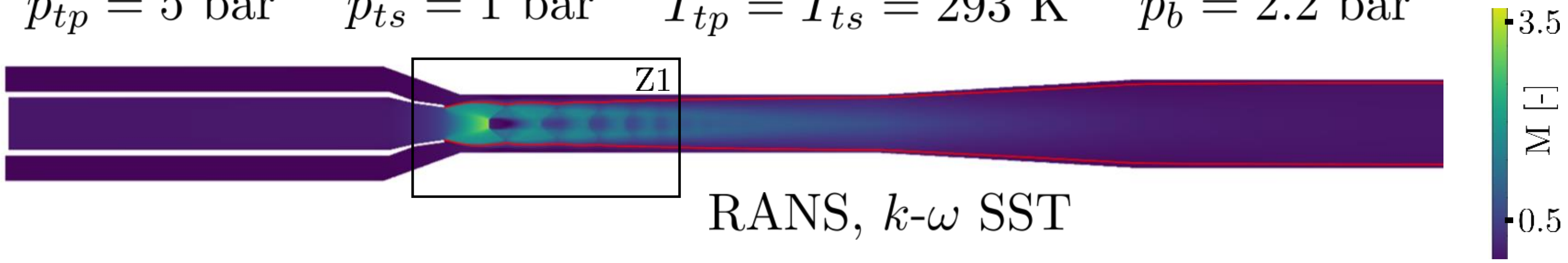
$$\begin{cases} d_x(\dot{m}_s) &= 0 \\ d_x(\dot{m}_s u_s) &= -A_s d_x p_s + \frac{1}{2} f_{ps} \left(\rho_p u_p^2 - \rho_s u_s^2\right) l_{ps} - \frac{1}{2} f_w (\rho_s u_s^2) l_w \\ d_x(\dot{m}_s h_{ts}) &= 0 \end{cases}$$

Shear

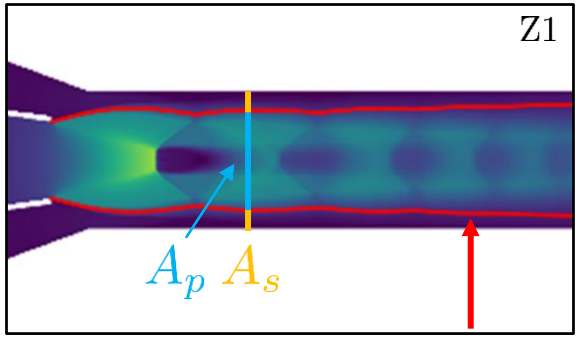
Wall friction

From 2D axisymmetric CFD to 1D reference data

$p_{tp} = 5$ bar $p_{ts} = 1$ bar $T_{tp} = T_{ts} = 293$ K $p_b = 2.2$ bar



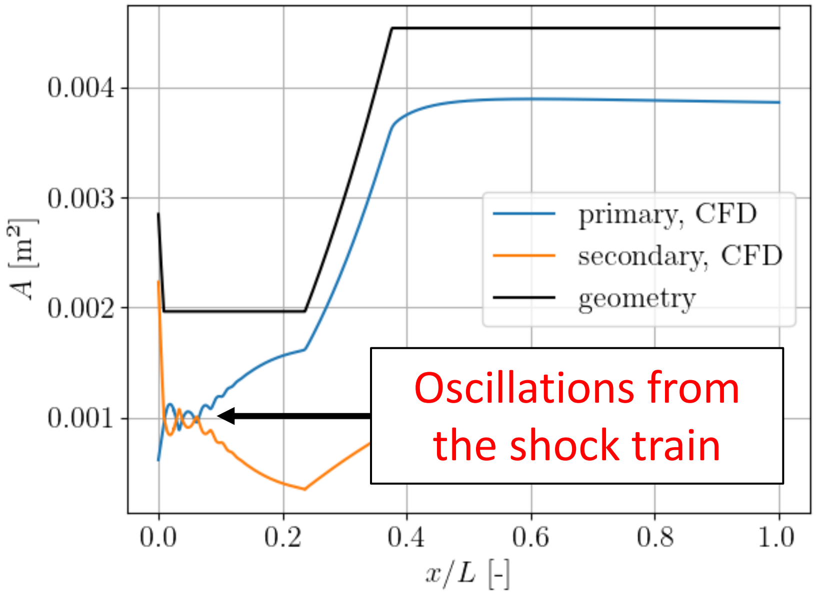
RANS, k-ω SST

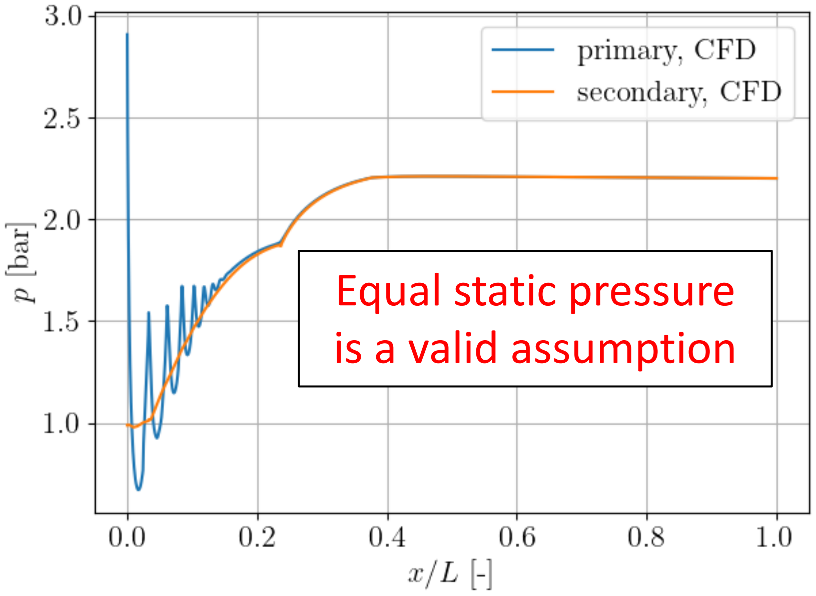


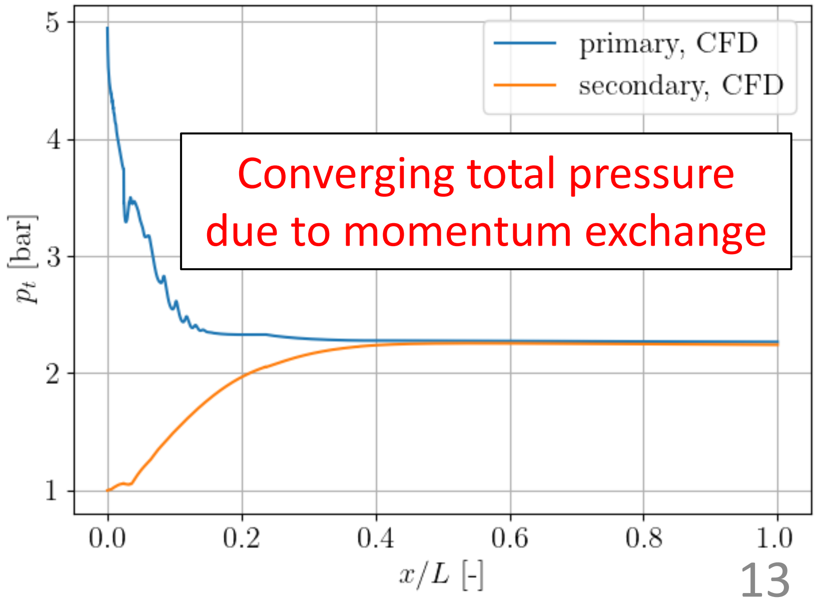
Dividing streamline

1D reference data through integration:

$$\tilde{\boldsymbol{u}}_p = \frac{1}{A_p}\int_{A_p} \boldsymbol{u}_p dA_p \qquad \tilde{\boldsymbol{u}}_s = \frac{1}{A_s}\int_{A_s} \boldsymbol{u}_s dA_s$$



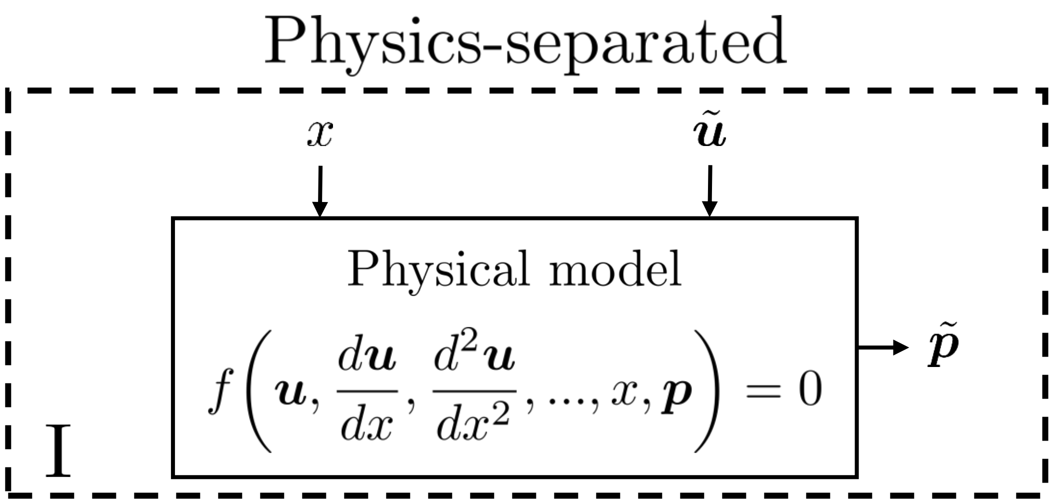




Contents

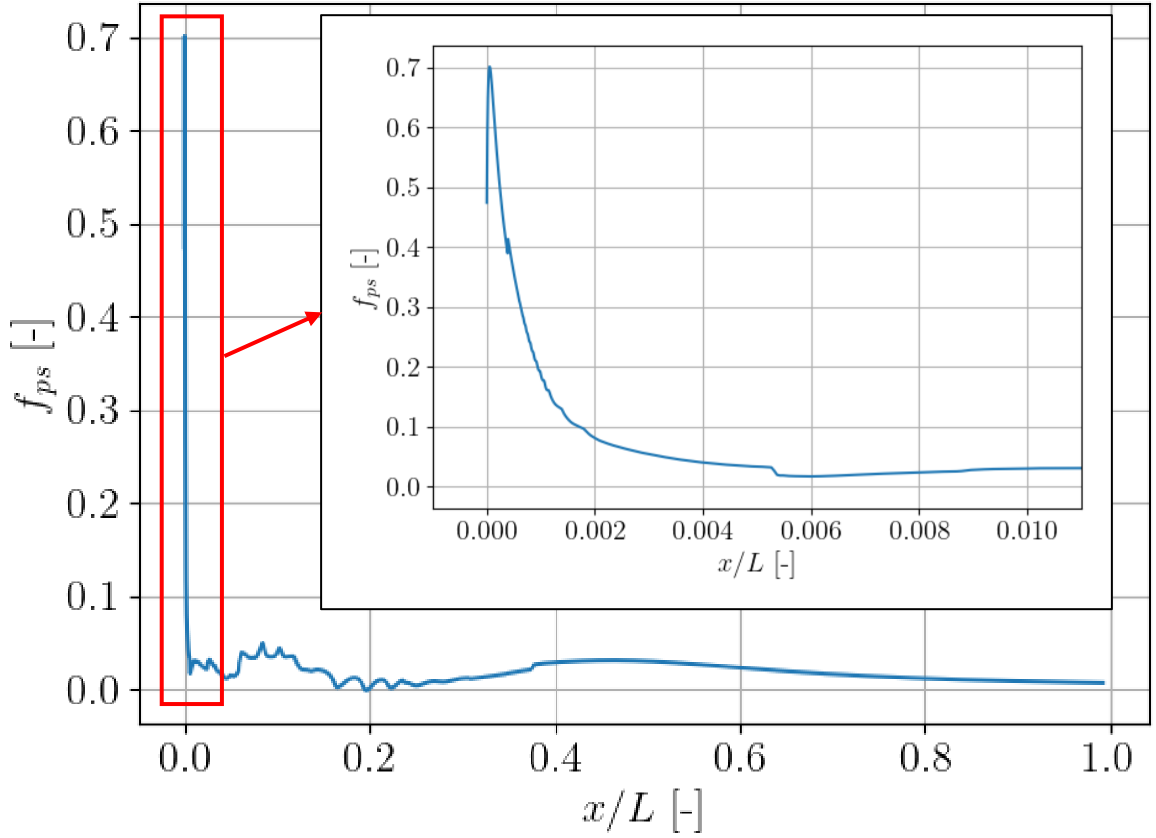
- A general framework for machine learning-based calibration
- **Application to a 1D ejector model**
  - Working equations & reference data
  - **Physics-separated approach**
  - Physics-integrated approach
- Conclusion

Results of the physics-separated approach (step I)



From the governing equations:

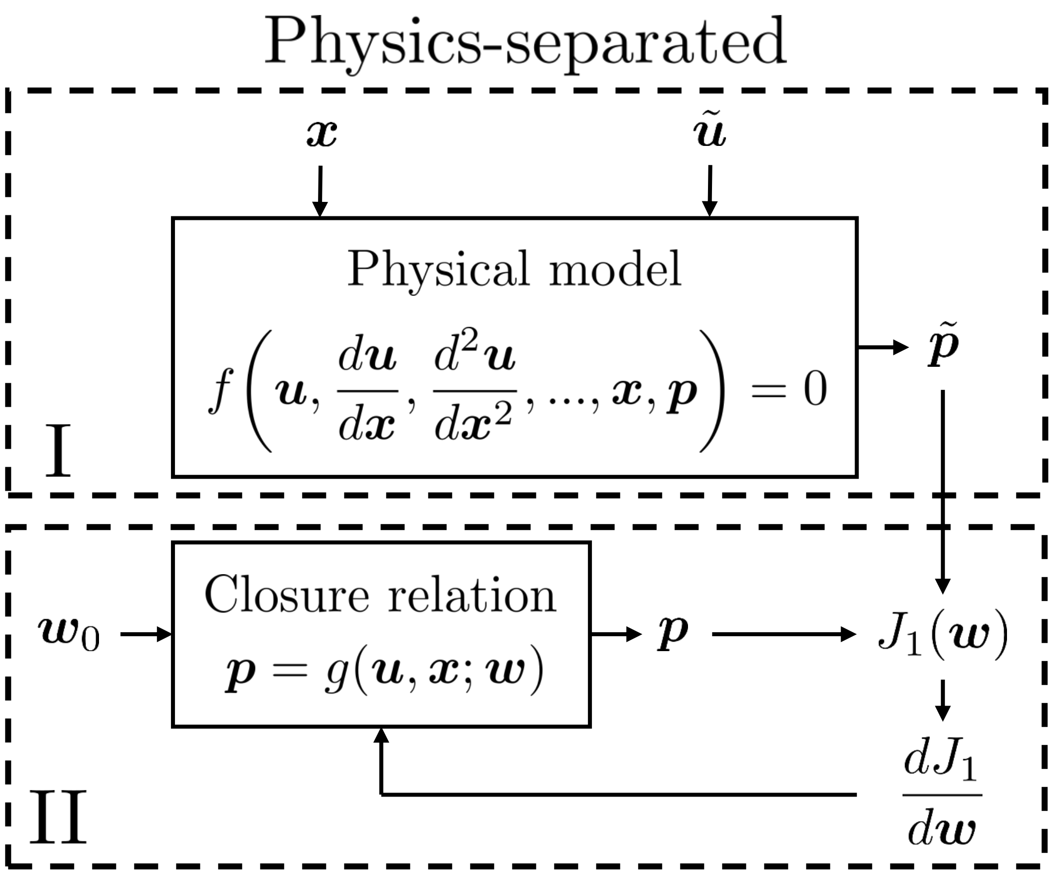
$$\underbrace{f_{ps} = -\frac{2A_p}{l_{ps}\gamma\left(M_p^2 - M_s^2\right)} \frac{d_x p_{tp}}{p_{tp}}}_{\text{from the post-processed CFD}}$$

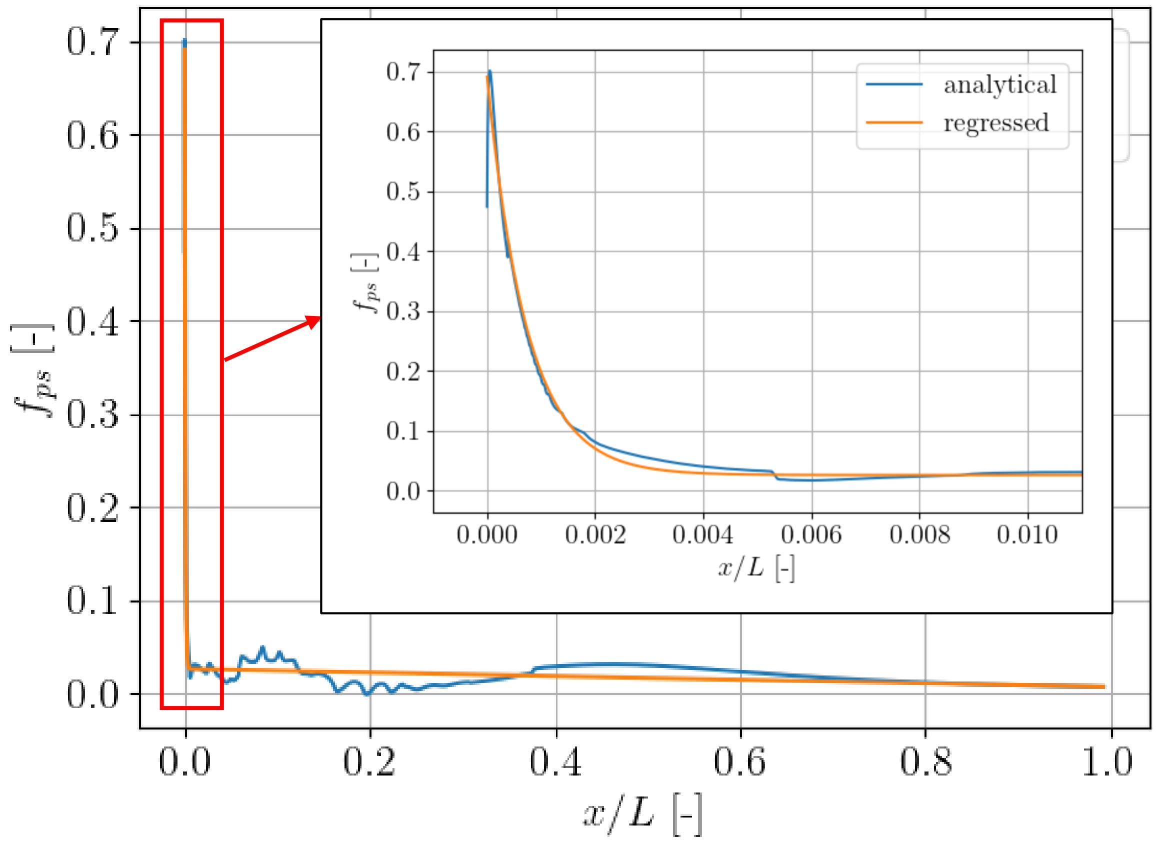


Candidate function:

$$\begin{cases} y = w_1 + w_2 \exp(-w_3 x) & \text{if } x \leq w_0 \\ y = y(w_0) + w_4 x & \text{if } x > w_0 \end{cases}$$

Results of the physics-separated approach (step II)

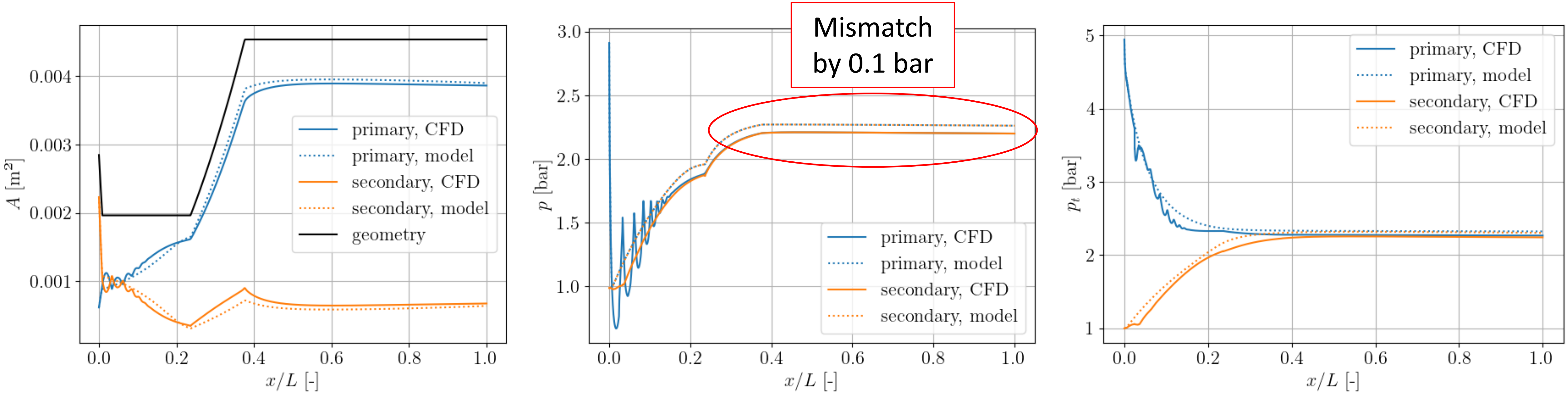




Candidate function:

$$\begin{cases} y = w_1 + w_2 \exp(-w_3 x) & \text{if } x \leq w_0 \\ y = y(w_0) + w_4 x & \text{if } x > w_0 \end{cases}$$

Results of the physics-separated approach (flow field)

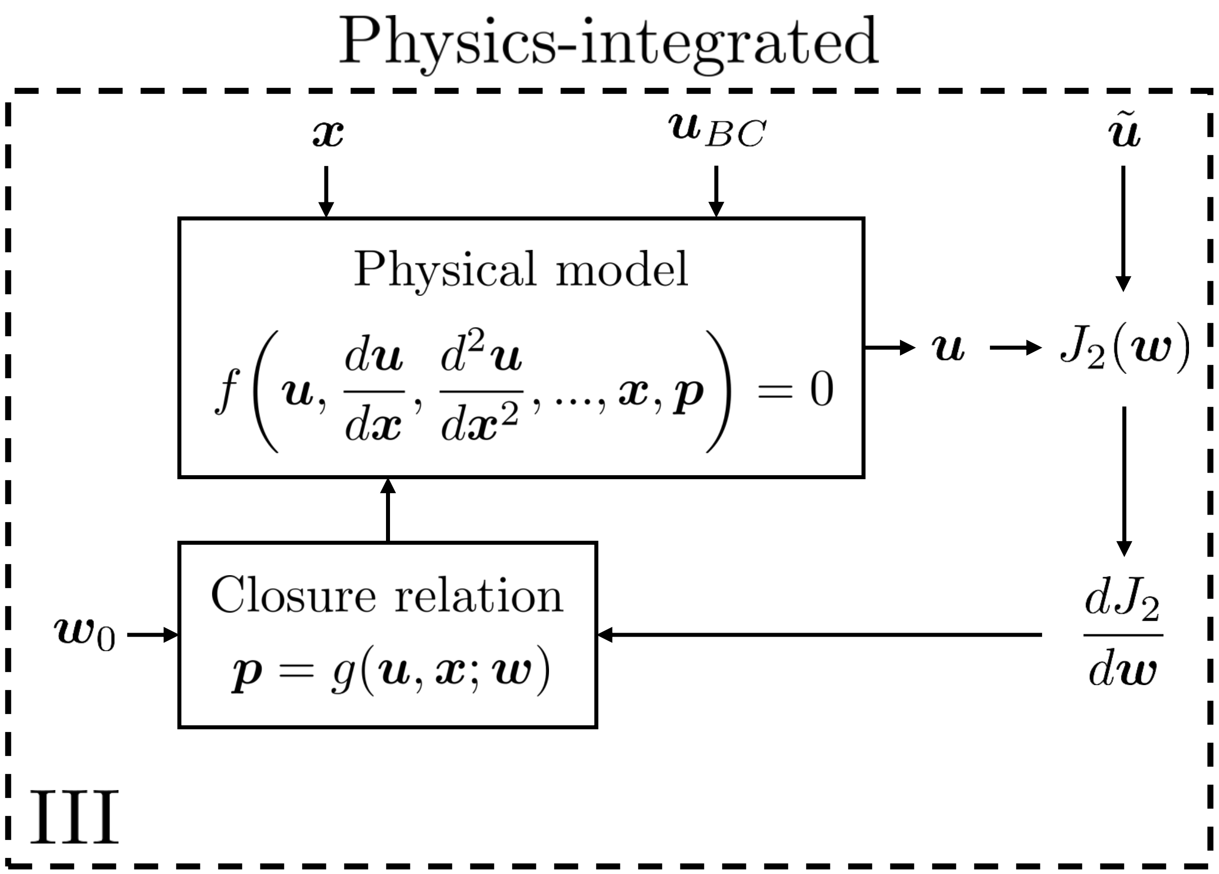


- ✓ The 1D model represents a cross-sectional average of the 2D / 3D flow field.
- ✓ The physics-separated approach provides a good initial calibration
- ✗ A mismatch on the state variables is possible (the training is unaware of the model).

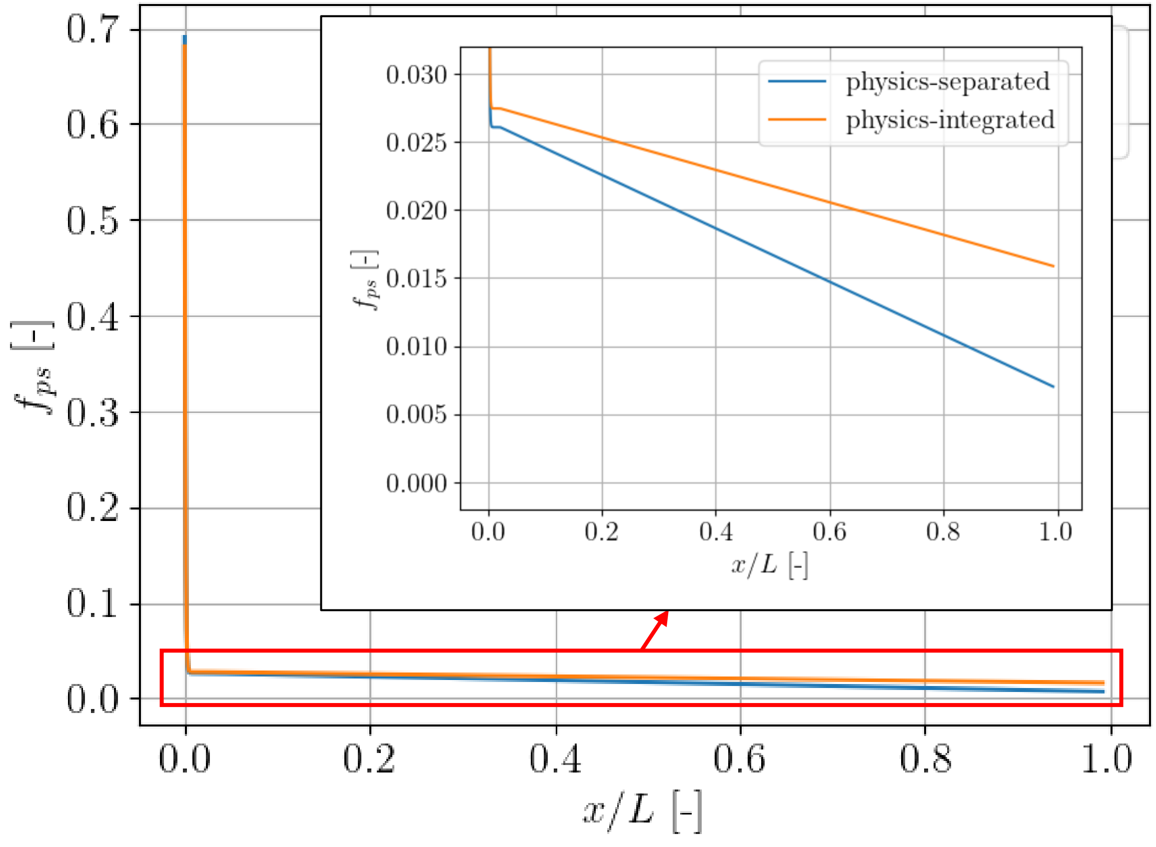
Contents

- A general framework for machine learning-based calibration
- **Application to a 1D ejector model**
  - Working equations & reference data
  - Physics-separated approach
  - **Physics-integrated approach**
- Conclusion

Results of the physics-integrated approach



The initial weights $\boldsymbol{w}_0$ come from the physics-separated approach

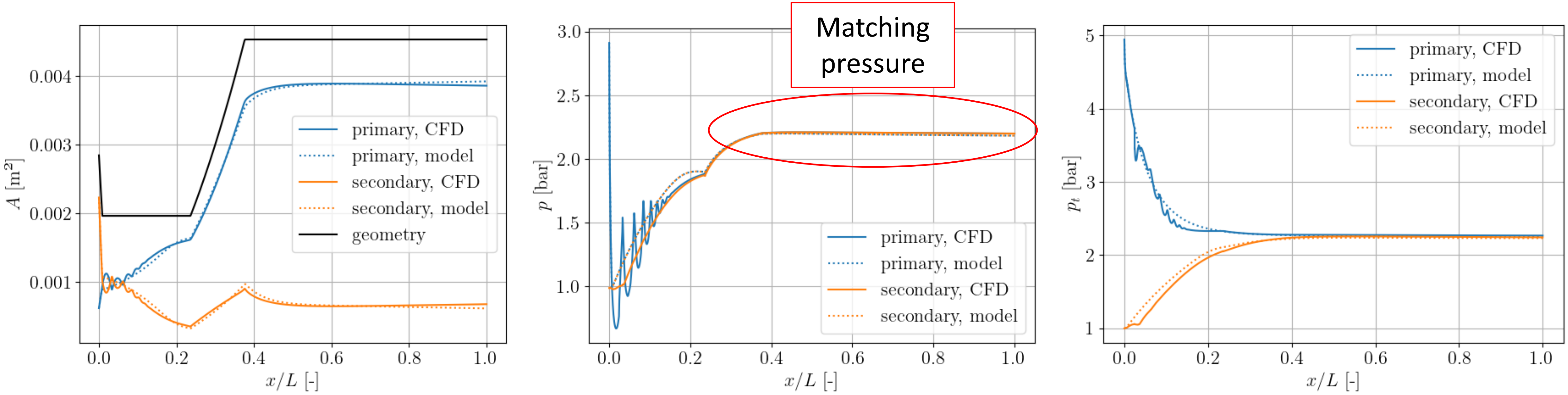


A difference by a factor two at the outlet!

Results of the physics-integrated approach (flow field)

Previous result (physics-separated)

New result (physics-integrated)

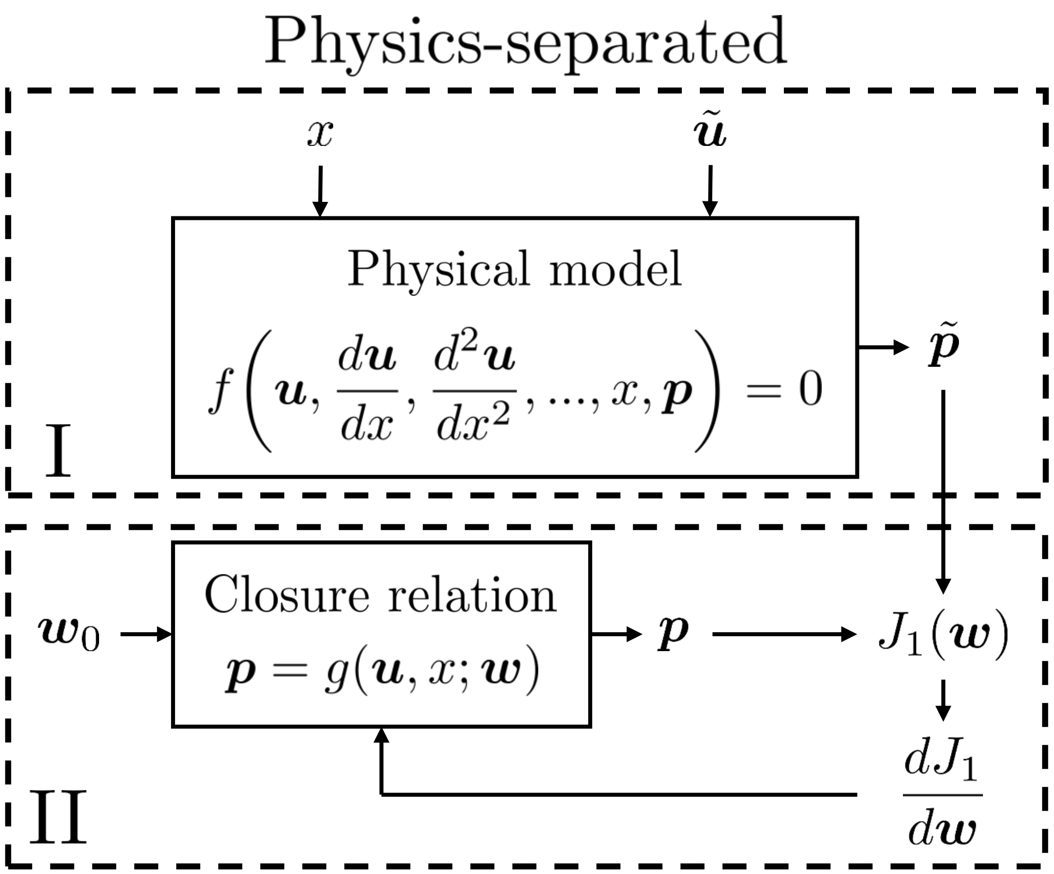


✓ The physics-integrated approach refined the previous calibration on all variables!

Contents

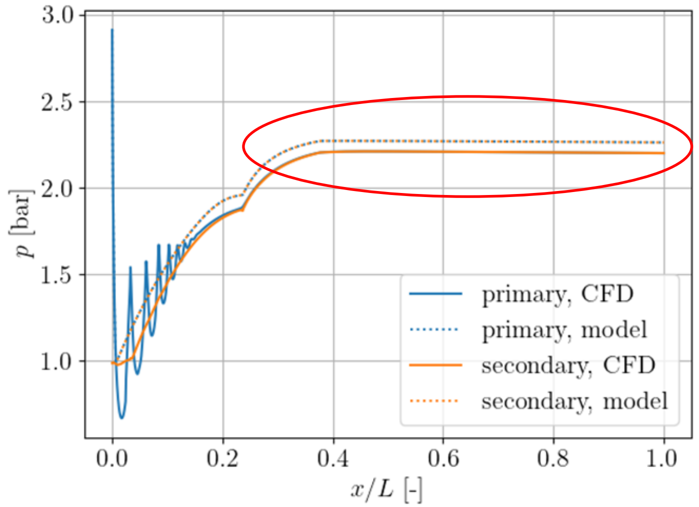
- A general framework for machine learning-based calibration
- Application to a 1D ejector model
- **Conclusion**

Physics-constrained machine learning for calibration



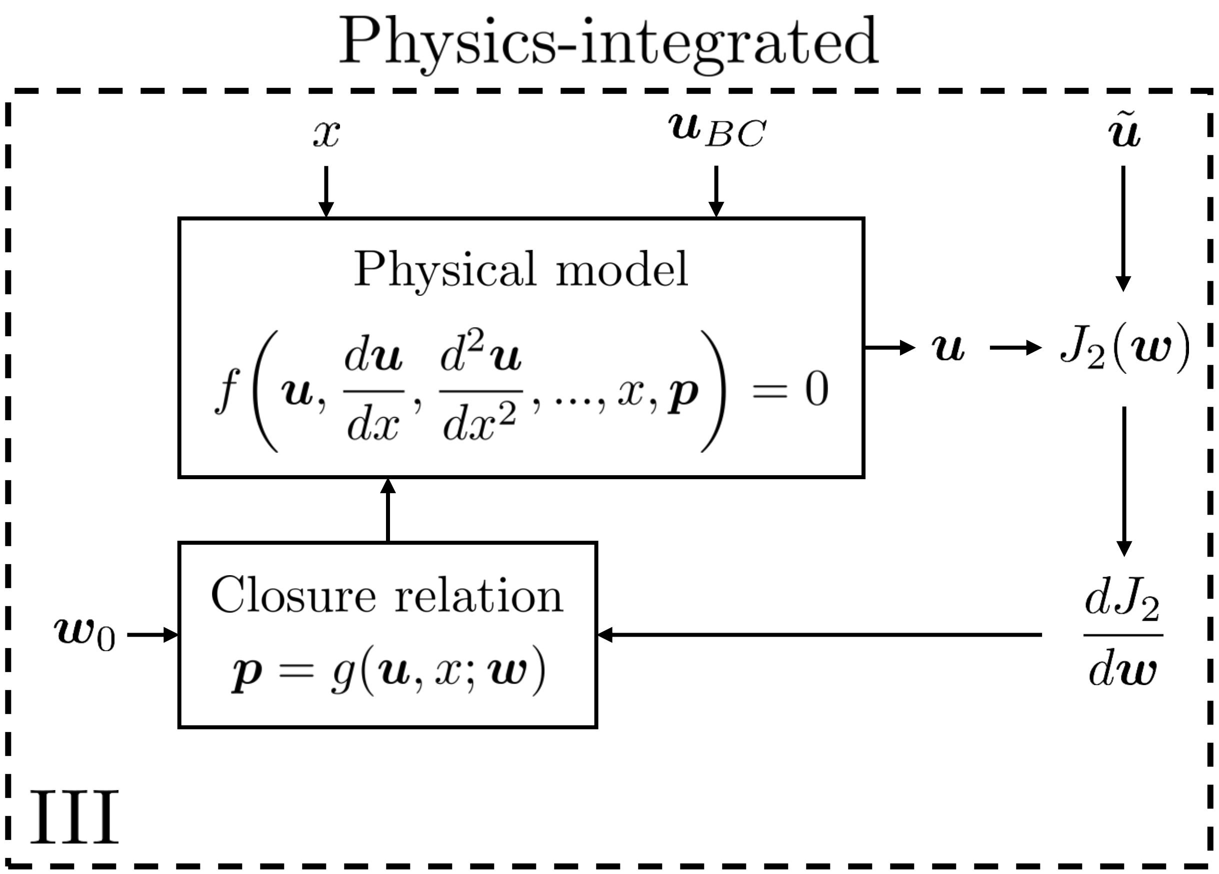
Physics-separated calibration

- ✓ Easy to train
- ✓ Guides the choice of the regressor
- ✗ Need dense data
- ✗ Unaware of the model



Mismatch by 0.1 bar

Physics-constrained machine learning for calibration



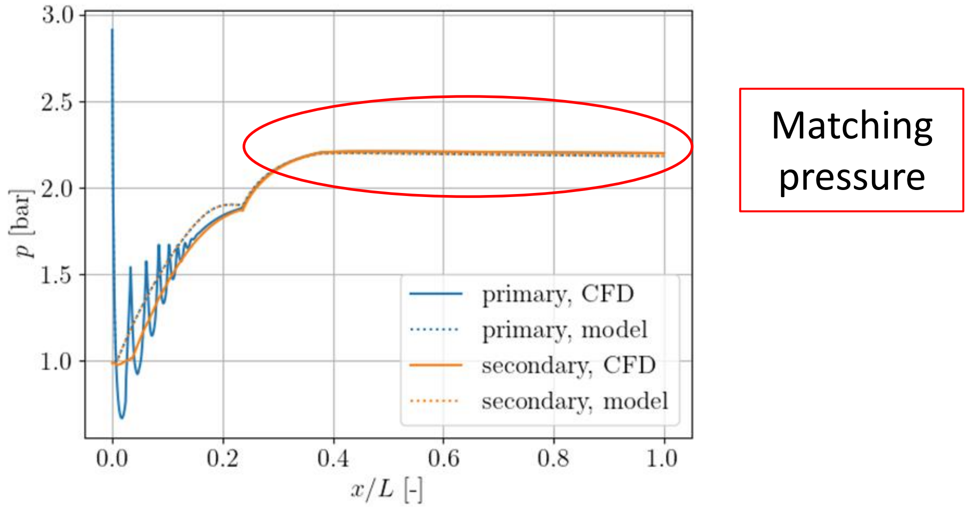
Physics-separated calibration: "first guess"

- ✓ Easy to train
- ✓ Guides the choice of the regressor
- ✗ Need dense data
- ✗ Unaware of the model

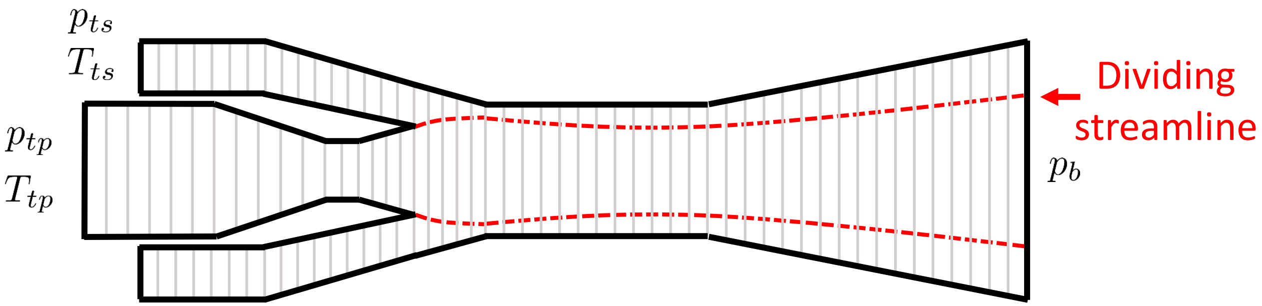
Physics-integrated calibration: "refinement"

- ✓ Aware of the model
- ✗ More complex to train

The combination of both is most convenient!

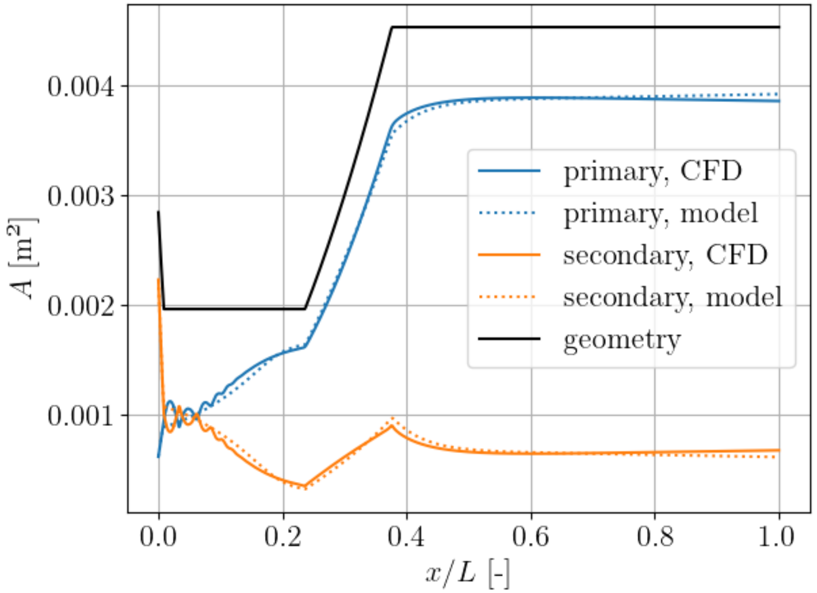


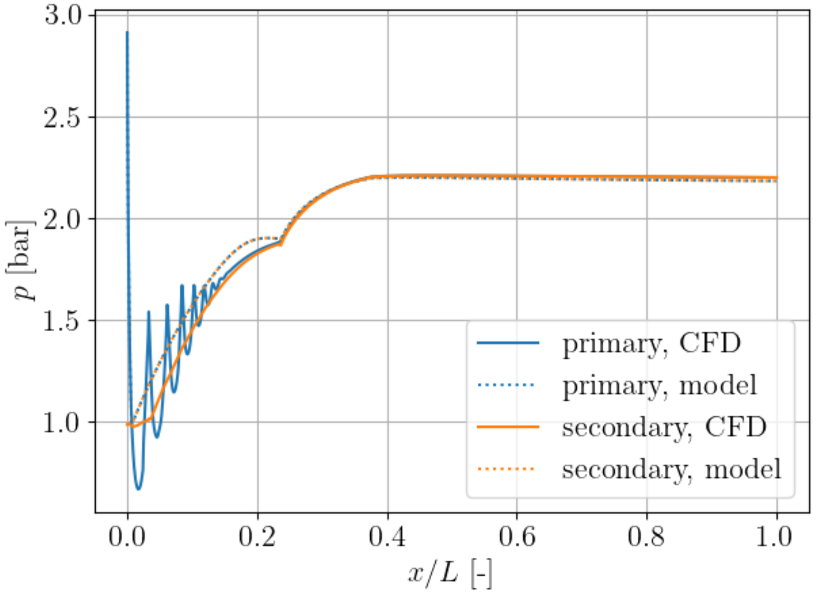
A 1D ejector model with two streams

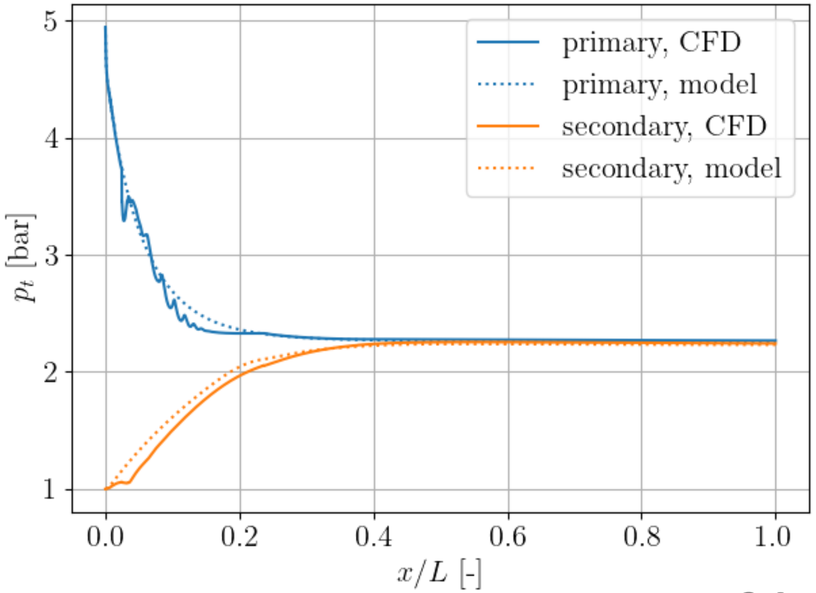


Conservation of mass, momentum and energy for each stream

- ✓ Highly interpretable
- ✓ Accurate versus area-averaged CFD
- ✓ Cheap (seconds versus days for CFD)
- ✗ Needs calibration

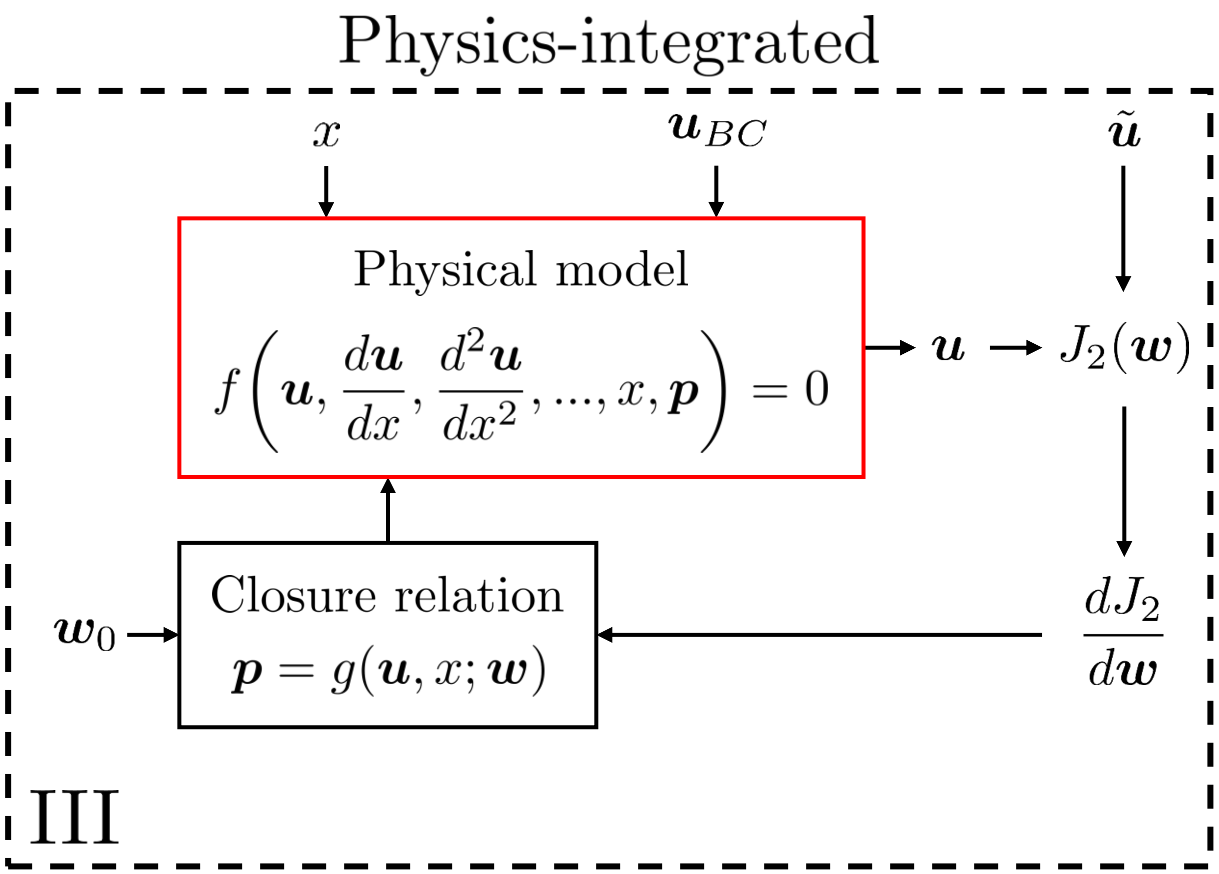






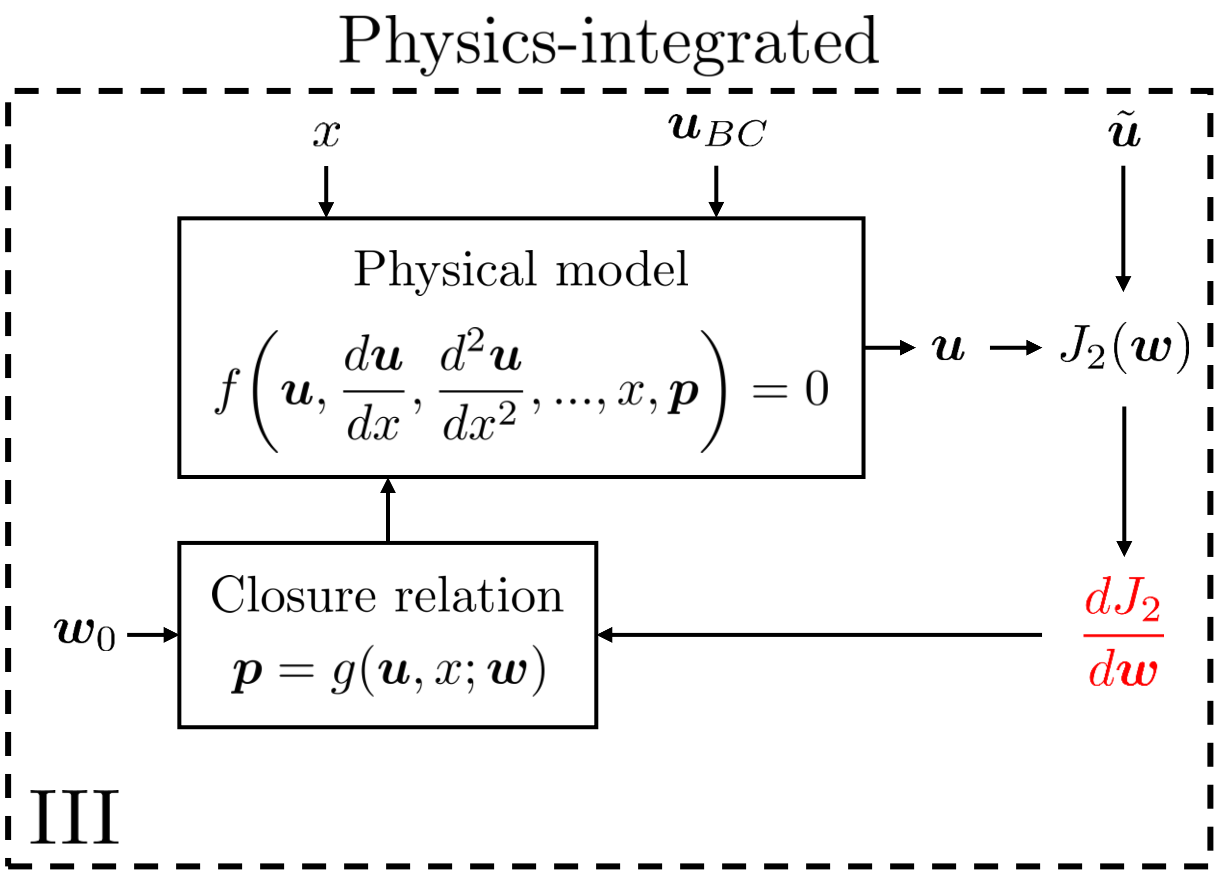
Accurate representation of area-averaged CFD

Ongoing work



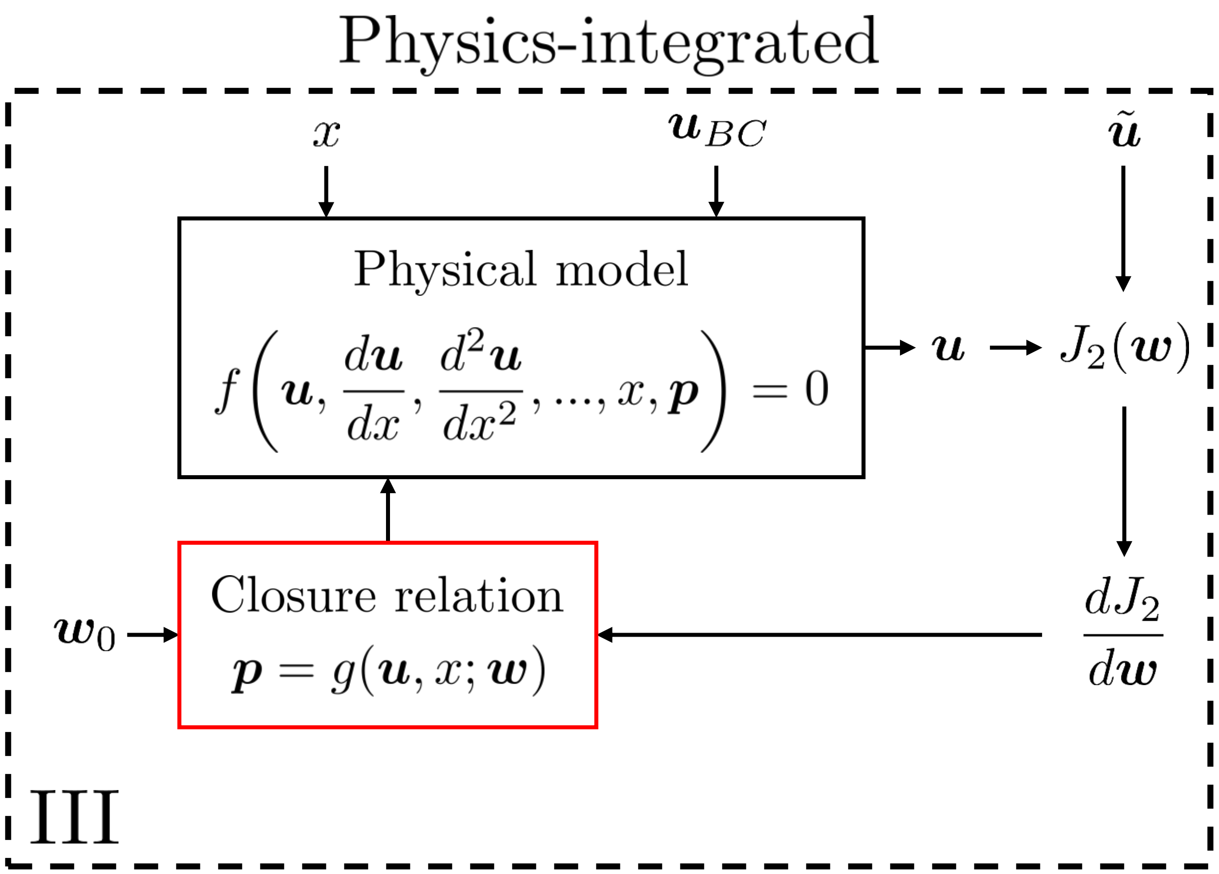
Extend the ejector model to handle choked flow

Ongoing work



- Extend the ejector model to handle choked flow
- Explore the adjoint method to compute the gradient

Ongoing work



Extend the ejector model to handle choked flow

Explore the adjoint method to compute the gradient

Link the shear coefficient to flow variables

$$\boldsymbol{p} = g(x) \longrightarrow \boldsymbol{p} = g(Re_p, Re_s, ...)$$

References

1. Parish E. J., Duraisamy K., A paradigm for data-driven predictive modeling using field inversion and machine learning. Journal of computational physics (2016).
2. Holland J. R., Baeder J. D., Duraisamy K., Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training. AIAA Aviation Forum (2019).
3. Sirignano J., MacArt J. F., Freund J. B., DPM: A deep learning PDE augmentation method with application to large-eddy simulation. Journal of Computational Physics (2020).
4. MacArt J. F., Sirignano J., Freund J. B., Embedded training of neural-network subgrid-scale turbulence models. Physical Review Fluids (2021).

A machine learning-based calibration of a 1D ejector model from CFD

36th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS)

25-30 June 2023

jan.vandenberghe@vki.ac.be

Jan Van den Berghe
Jagadish Babu Vemula
Yann Bartosiewicz
Miguel Alfonso Mendez